(m,n)-mer - A Simple and New Statistical Feature for viral Classification

Amanda Araújo Serrão de Andrade[1]***, Marco Grivet***[2]***, Otávio Brustolini***[1]***, and Ana Tereza Ribeiro de Vasconcelos*** [*1]

[1]National Laboratory for Scientific Computing, Bioinformatics Laboratory (LABINFO), Petrópolis, Rio de Janeiro, Brazil
[2]Pontifícia Universidade Católica do Rio de Janeiro, 22451-900, Brazil

[*]atrv@lncc.br

October 13, 2022

Abstract

The (m,n)-mer is a new statistical feature based upon conditional frequencies (conditional probability density distributions). Here, we present the mnmer function and show a pratical example of classification using mnmer output.

Package

mnmer 0.1.0

Contents

1 Introduction to the (m,n)-mer concept

The (m,n)-mer R package was created to summarize biological data into numerical characteristics, as an alternative for k-mers. It reads a FASTA file and generates a table describing the conditional frequency distribution of the selected (m,n)-mer in the sequences. This output is combined with class information to generate the feature matrix for classification.

Since letters are a bit awkward from a mathematical viewpoint, lets univocally associate the digits 0, 1, 2 and 3 to letters A, C, G and T. Any order will do. Hence, each k-mer can be described by a unique base-4 integer number in the range from 0 to $4.k - 1$. As an example, consider the 6-mer ACCTGA and the association described above. Then the 6-mer ACCTGA can be represented as the base-4 number 011320, which corresponds to the number 376 in the decimal notation.

If we order all the k-mers according to these numbers, we say the k-mers are "lexicographically" ordered and we can use the notation s^i_k to identify the k-mer associated to the decimal number i ranging from 0 to $4.k - 1$. Following the above example, we can say that $s^{376}_6 = ACCTGA$. Let's now consider a particular genome extracted from a particular organism and assume that we count the occurrence of all distinct k-mers present in this genome. We will denote by c^i_k the count corresponding to the i-th k-mer. Then we can create a real vector of size $4.k$ as illustrated in the Figure 1 below:

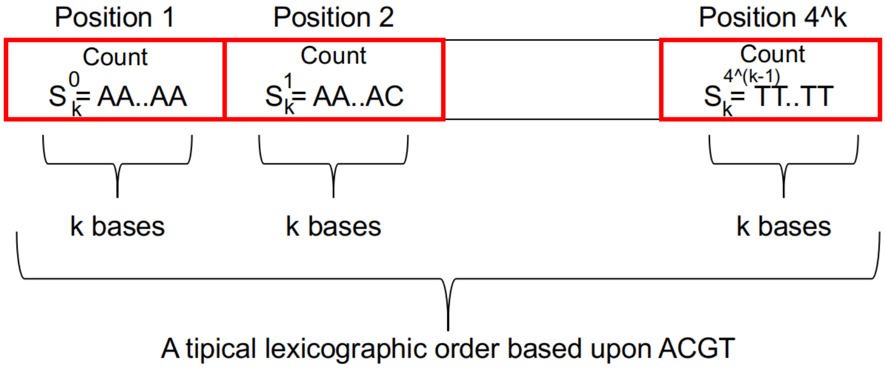


Figure 1: k-mer organization for k-mer and (m,n)-mer statistical distributions evaluation

Since this counting is generally made for several organisms with different sizes, for the sake of comparison it is more convenient express this count in relative terms. Similarly, we will denote by as f^i_k the relative frequency of k-mer s^i_k which is computed as the division $f^i_k = \frac{c^i_k}{N_k}$ where N_k is the total number of k-mers counted in this organism, that is, $N_k = \sum_{i=0}^{4^k-1} c^i_k$.

Let's define a vector $\underline{f}_k = (f^0_k, f^1_k, ..., f^{4^k-1}_k)$ as the formal descriptor of the particular organism as far as k-mer is concerned.

Please notice that each element of vector $\underline{f}_k$ is nonnegative and their sum is 1, which allow us to interpret this vector as a "probability density distribution" according to statistical parlance.

In the above example, consider the 6-mer ACCTGA and m and n respectively assuming the values 4 and 2. Hence we have $S_6 = ACCTGA$, $S^-_4 = ACCT$ and $S^+_2 = GA$. Although we have used the superscripts $-$ and $+$ to respectively indicate the left and right part of the k-mer S_k, they are also an m-mer and an n-mer on its own. We propose the replacement of the unconditional frequency f^i_k by the conditional frequency $f^i_{m,n}$ which represents the relative frequency of the n-mer S^+_n conditioned to the fact that the set of m bases that precedes it is S_m^-. The vector so defined will be conveniently renamed $\underline{f}_{m,n}$ in order to be more explicit. By defining u/v as the result of the integer division of u by v then, based upon the classical result of conditional probabilities, we have $\underline{f}_{m,n} = (f^0_{m,n}, f^1_{m,n}, ..., f^{4^k-1}_{m,n})$ where $f^i_{m,n} = \frac{f^{i]}_k}{f^{i/4^n}_m}$.

Again in our example, f_6^{376}, the relative frequency of 6-mer ACCTGA must be divided by f_4^{23}, which is the relative frequency of $S_4^{23} = ACCT$.

It can be easily seen that the sum all elements of this new vector is no longer 1 but $4m$, because it is the concatenation of $4m$ conditional frequency distributions associated to each one of the possible m-mers. This fact has no impact whatsoever in our current discussion but, in order to keep it as a probability density distribution as well, we normalize it by dividing it by $4m$. Figure 2 shows an comparison of k-mers and (m,n)-mers obtained from the same nucleotide sequence.

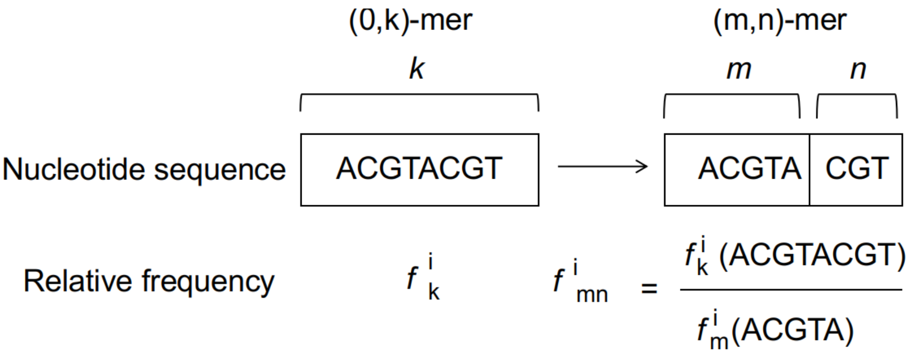


Figure 2: Comparing k-mer to mn-mer relative frequency

According to Figure 3 below, the k-mers are represented as (0,k) and the (m,n)-mers as (m,n).

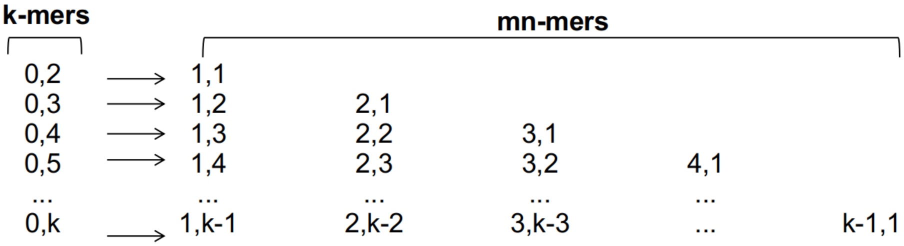
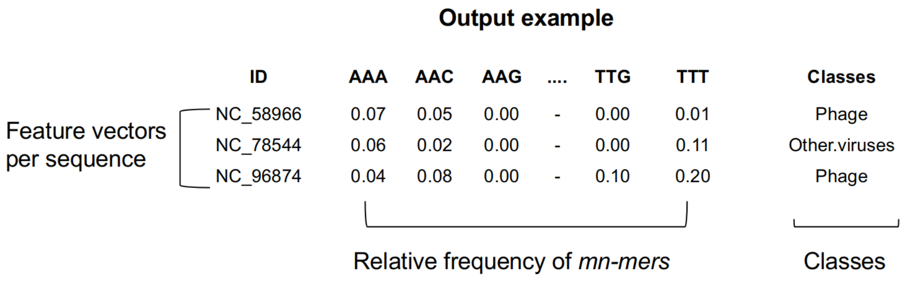


Figure 3: Numeric representation

The output table (Figure 4) includes the fasta file accession numbers as an ID column, the relative frequency of mn-mers up to $4.k$ columns, and class information.

For more details and performance comparison, please see Andrade et al., 2022 (in press).



ID	AAA	AAC	AAG	....	TTG	TTT	Classes
NC_58966	0.07	0.05	0.00	-	0.00	0.01	Phage
NC_78544	0.06	0.02	0.00	-	0.00	0.11	Other.viruses
NC_96874	0.04	0.08	0.00	-	0.10	0.20	Phage

Figure 4: Output example

2 The (m,n)-mer as an R package

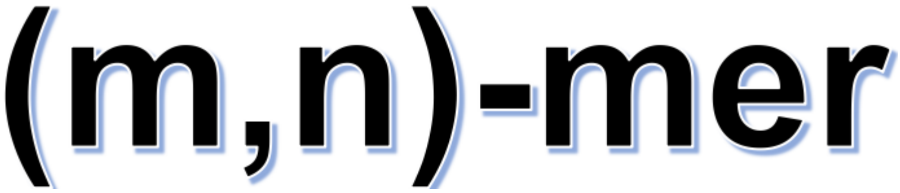


Figure 5: Package logo

The *mnmer* R package was created to summarize biological data into numerical characteristics. It takes a FASTA file and produces a dataframe describing the relative frequency of all (m,n)-mers found in the input sequences. This output is combined with class information to generate a feature matrix for classification.

Limitations:

In this first version, the package only supports plain FASTA files (no compression required); All bases different from A, C, T, and G are ignored in the (m,n)-mer generation; No paralelism or specific treatment for big FASTA; The user needs to add the data classes prior to classification;

2.1 Dependencies

The package only needs *R* 4.0.0 or later.

2.2 Instalation

The user should install the package from the GitHub repository. It can be done by using the *devtools* package.

```
library(devtools)
install_github("labinfo-lncc/mnmer", ref="main")
```

2.3 The mnmer function

The ‘mnmer‘ function is the main function of this package. It generates the dataframes containing the conditional probability. This function can generate both k-mers and (m,n)-mers, by calling the function ‘cmmer‘ from the C++ script.

```
dataframe <- mnmer(file, k, m)
```

The parameters receives:

file = FASTA file (it could be an multiFASTA)

k = Value of k for k-mer generation. Needs to be different from zero.

m = Value of m for (m,n)-mer generation in the format of (m, k-m). In case of k-mer generation, m should be zero as (0,k).

2.4 Pratical example

Assume we need to distinguish between viruses detected in mosquito samples and viruses that exclusively infect plants. After instalation, the user should run:

```
library("mnmer")
dir <-system.file("extdata", package="mnmer")
```

2.4.1 Produce k-mer distributions

The parameter k is set to choice for k-mer generation, while the parameter m is set to zero. Considering that the k-mers are conditioned to zero bases.

```
mosquito <- mnmer(file.path(dir, "mosquito_vir.fasta"),2,0)
plant <- mnmer(file.path(dir, "plant_vir.fasta"),2,0)
```

2.4.2 Produce (m,n)-mer distributions

The k and m parameters are chosen by the user for mn-mer creation. For instance, k = 2 and m = 1 yield the (1,1)-mer, in which one base is conditioned on the frequency of one preceding base.

```
mosquito <- mnmer(file.path(dir, "mosquito_vir.fasta"),2,1)
plant <- mnmer(file.path(dir, "plant_vir.fasta"),2,1)
```

Bases other than A, C, T, and G were disregarded.

For classification outside of the mnmer program, we utilize the (1,1)-mer feature matrices. Here's a real-world example of code using *Caret*, *MLeval* and *data.table*:

```
library(data.table)
library(caret)
library(MLeval)

classes <- replicate(nrow(mosquito), "mosquito.vir")
featureMatrix_mosquito <- cbind(mosquito,classes)
classes <- replicate(nrow(plant), "plant.vir")
featureMatrix_plant <- cbind(plant,classes)

featureMatrix <- rbind(featureMatrix_mosquito, featureMatrix_plant)
featureMatrix <- subset(featureMatrix, select = -c(seqid))
train_index <- createDataPartition(featureMatrix$classes, p=0.8, list=FALSE)
```

```
train <- featureMatrix[train_index, ]
test <- featureMatrix[-train_index, ]
control <- trainControl(method="cv",
                        summaryFunction=twoClassSummary,
                        classProbs=T,
                        savePredictions = T)
roc <- train(classes ~ .,
            data=train,
            method="rf",
            preProc=c("center"),
            trControl=control)
res <- evalm(roc) # Make the ROC plot
```

This classification produces the ROC curve below:

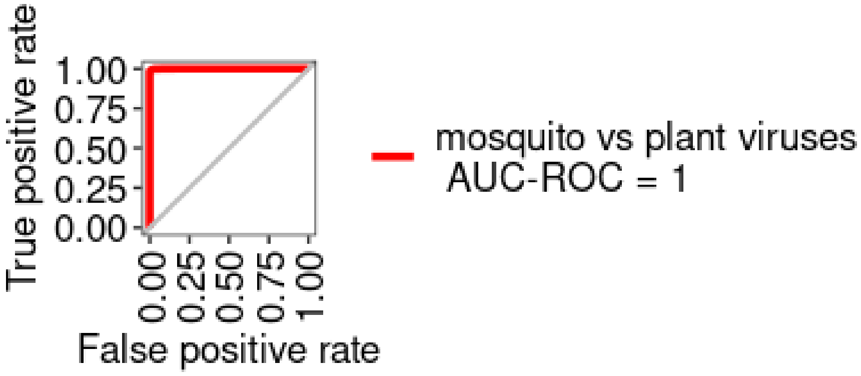


Figure 6: ROC curve and Area Under the Curve Value

2.5 What to expect from newest versions

We would like to add new functionalities related to handle big multiFASTA files, paralelism and compressed files. We are also implementing the option to generate the (m,n)-mer distributions for IUPAC and N bases as well.

2.6 SessionInfo

```
utils::sessionInfo()

## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=pt_BR.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=pt_BR.UTF-8        LC_COLLATE=pt_BR.UTF-8
##  [5] LC_MONETARY=pt_BR.UTF-8    LC_MESSAGES=pt_BR.UTF-8
##  [7] LC_PAPER=pt_BR.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=pt_BR.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] digest_0.6.29       magrittr_2.0.3      evaluate_0.17
##  [4] highr_0.9           rlang_1.0.6         stringi_1.7.8
##  [7] cli_3.4.1           rmarkdown_2.17      BiocStyle_2.25.0
## [10] tools_4.2.1         stringr_1.4.1       xfun_0.33
## [13] yaml_2.3.5          fastmap_1.1.0       compiler_4.2.1
## [16] BiocManager_1.30.18 htmltools_0.5.3     knitr_1.40
```